Steady State Analysis of Random Early Detection Gateway with Controlled Traffic by TCP

— Extended Abstract —

Hiroyuki Ohsaki, Masayuki Murata, and Hideo Miyahara

Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

(Phone) +81-6-6850-6588
(Fax) +81-6-6850-6589
(E-mail) oosaki@ics.es.osaka-u.ac.jp

1 Introduction

In a packet-switched network, a feedback-based congestion control mechanism is essential to provide data transfer services efficiently. Its main objective is to prevent packet losses in the network, and to utilize network resources effectively. The current Internet uses a window-based flow control mechanism in its TCP (Transmission Control Protocol), as the feedback-based congestion control mechanism. As an example, a version of TCP mechanism called *TCP Reno* uses packet losses in the network as feedback information from the network since packet loss implies congestion occurrence in the network [1, 2]. Until packet loss occurs in the network, TCP Reno gradually increases its window size. As the window size exceeds its available bandwidth, excess packets are absorbed by the buffer of an intermediate gateway for some period. If the window size increases further, the buffer of the gateway overflows, leading to packet losses. The source host detects an occurrence of packet losses in the network by, for example, receiving duplicate ACKs, and quickly reduces its window size. After reduction of the window size, congestion in the network will be relieved. Once congestion in the network is over, the source host increases its window size again. In short, the congestion control mechanism of TCP first increases its window size, and as soon as it detects packet losses in the network, it reduces its window size. The congestion control mechanism of TCP repeats this process indefinitely during the connection.

The congestion control mechanism of TCP was designed to work without any knowledge on underlying network including the gateway's algorithm. Namely, the congestion control mechanism of TCP assumes nothing about a gateway's operation. It is because neither a packet scheduling discipline nor a packet discarding algorithm of the gateway is known by the source host in real networks. Actually, separation of TCP's congestion control mechanism from the gateway's algorithm is desirable when several types of congestion control mechanisms and gateway's algorithms co-exist in the network as in the current Internet. However, such a generality of the congestion control mechanism of TCP significantly limits the network performance.

Accordingly, several gateway-based congestion control mechanisms have been proposed to support an end-to-end congestion control mechanism of TCP [3, 4, 5]. One of promising gateway-based congestion control mechanisms is a RED (Random Early Detection) gateway [4]. The key idea of the RED gateway is to keep the average queue length (i.e., the average number of packets in the buffer) low. Basically, the RED gateway randomly discards an incoming packet with a probability that is determined based on the average queue length. The operation algorithm of the RED gateway is quite simple so that the RED algorithm can be easily implemented in real gateways. The authors of [4] have claimed advantages of the RED gateway over a conventional Tail-Drop gateway as follows: (1) the average queue length is kept low, so that an end-to-end delay of a TCP connection is also kept small, (2) the RED gateway has no bias against bursty traffic as in the Tail-Drop gateway, and (3) a global synchronization problem of TCP connections found in the Tail-Drop gateway is avoided. However, the characteristics of the RED gateway has not been fully investigated.

There have been several simulation studies of the RED gateway [4, 6, 7]. But there still remain open issues. The hardest one is how to determine several control parameters of the RED gateway. Despite the fact that effectiveness of the RED gateway is fully dependent on a choice of control parameters [6, 8, 9], it has not been known how to configure those control parameters. The authors of [4] have proposed a recommended set of control parameters, but it is just an empirical guideline without any theoretical basis. There are only a few analytical studies on the RED gateway. In [10], the authors have analyzed the performance of the RED gateway for bursty and less bursty traffic. However, their analytic model is very simple and not realistic; the input traffic to the RED gateway is modeled by a batch Poisson process, which completely neglects the dynamics of the congestion control mechanism of TCP. In [11], the authors have analyzed the dynamics of the RED gateway, but the input traffic is still limited to either a renewal process or an MMPP (Markov Modulated Poisson Process). Since the RED gateway was designed to cooperate with the congestion control mechanism of TCP, dynamics of TCP must be taken into account to understand the substantial property of the RED gateway.

The objective of this paper is to analyze the behavior of the RED gateway under controlled traffic by TCP. We explicitly model the dynamics of the congestion control mechanism of TCP, and analyze the steady state behavior of the RED gateway. Through several numerical examples, we investigate how control parameters of the RED gateway affects its behavior. Our analytic result makes it possible to configure control parameters of the RED gateway under various network configurations.

This paper is organized as follows. In Section 2, the network model we will use throughout this paper is described, followed by brief explanation of the RED gateway. Section 3 is the main part of this paper, where steady state analysis of the RED gateway with controlled traffic by TCP is performed. In Section 4, we present several numerical examples to illustrate how control parameters of the RED gateway affects its behavior, and to demonstrate the validity of our analysis.

2 Analytic Model

Figure 1 illustrates our network model that will be used throughout this paper. It consists of a single RED gateway and the number N of TCP connections. All TCP connections have identical two-way propagation delays, which are denoted by τ [ms]. We assume that the processing speed of the gateway, which is denoted by B [packet/ms], is the bottleneck of the network. Namely, transmission speeds of all host–gateway links are assumed to be faster than the processing speed of the gateway.

We model the congestion control mechanism of TCP version Reno [2, 12] at all source hosts. A source host has a window size, which is controlled by the congestion control mechanism of TCP. By letting w be the current window size of a source host, it is allowed to send the number w of packets without receipt of corresponding ACK (Acknowledgement) packets. In other words, the source can send a bunch of w packets during a round-trip time. We then model the entire network as a discrete-time system, where a time slot of the system corresponds to the round-trip time of TCP connections. We define $w_n(k)$ [packet] as the window size of the source host n ($1 \le n \le N$) at slot k. All source hosts are assumed to have enough data to transmit; that is, the source host n is assumed to always send the number $w_n(k)$ of packets during slot k.

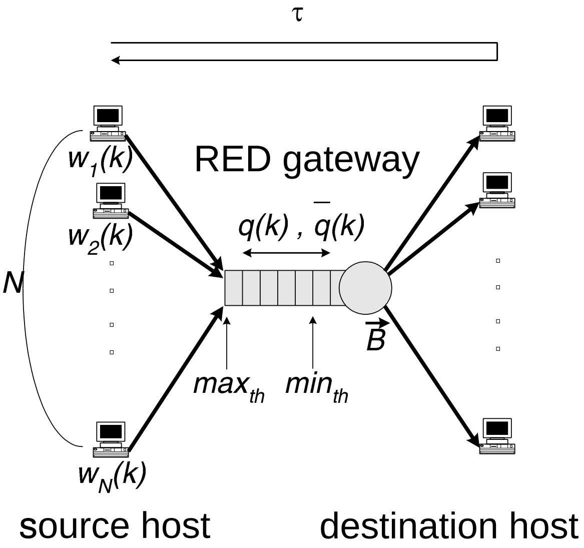


Figure 1: Analytic model.

The RED gateway has several control parameters. Let min_{th} and max_{th} be the *minimum* threshold and the *maximum* threshold of the RED gateway, respectively. These threshold values are used to determine a packet dropping probability for every incoming packet. The RED gateway maintains the average queue length (i.e., the average number of packets waiting in the buffer). The RED gateway uses an exponential weighted moving average (EWMA), which is a sort of low-pass filters, to calculate the average queue length from the current queue length. More specifically, let q and $\overline{q}$ be the current queue length and the average queue length. At every packet arrival, the RED gateway updates the average queue length $\overline{q}$ as

$$\overline{q} \quad \leftarrow \quad (1 - w_q)\, \overline{q} + w_q\, q, \tag{1}$$

where w_q is a weight factor. We define $q(k)$ [packet] and $\overline{q}(k)$ [packet/ms] be the current and the average queue lengths at slot k, respectively. We assume that both q and $\overline{q}$ are fixed within a slot. As discussed in [10], this assumption is realistic for a small value of w_q.

Using the average queue length, the RED gateway calculates the packet marking probability p_b at every arrival of an incoming packet. Namely, the RED gateway calculates p_b as

$$p_b = \begin{cases} 0 & \text{if } \overline{q} < min_{th}, \\ 1 & \text{if } \overline{q} \geq max_{th}, \\ max_p \left(\frac{\overline{q} - min_{th}}{max_{th} - min_{th}} \right) & \text{otherwise,} \end{cases} \quad (2)$$

where max_p is another control parameter that determines the maximum packet marking probability (Fig. 2). The packet marking mechanism of the RED gateway is not per-flow basis, so that the same packet marking probability is used for all TCP connections. A typical setting of control parameters of the RED gateway, which has been recommended by the authors of [4], is listed in Table 1. Refer to [4] for the detailed algorithm of the RED gateway.

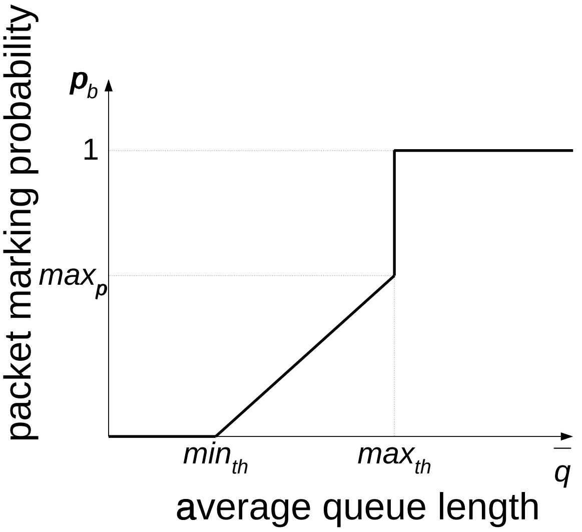


Figure 2: Relation between packet marking probability p_b and average queue length $\overline{q}$.

Table 1: Recommended set of RED control parameters

min_{th}	minimum threshold value	5
max_{th}	maximum threshold value	15
max_p	maximum packet dropping probability	0.1
w_q	weight factor for averaging	0.002

3 Steady State Analysis

3.1 Derivation of State Transition Equations

We first obtain the packet dropping probability in every slot, followed by discussions on its properties. At the beginning of slot k, the source host n sends the number $w_n(k)$ of packets into the network. The RED gateway marks each arriving packet based on the average queue length. As discussed in Section 2, the average queue length $\overline{q}$ is assumed to be fixed within a slot in our analysis. The packet marking probability is then fixed, and it is denoted by $p_b(k)$ at slot k. From Eq. (2), $p_b(k)$ is given by

$$p_b(k) = max_p \left(\frac{\overline{q}(k) - min_{th}}{max_{th} - min_{th}} \right). \quad (3)$$

Then, every arriving packet at the RED gateway is discarded with the probability of

$$\frac{p_b(k)}{1 - count \cdot p_b(k)}, \quad (4)$$

where *count* is the number of unmarked packets that have arrived since the last marked packet. So the number of unmarked packets between two consecutive marked packets can be represented by an uniform random variable in $\{1, 2, \cdots, 1/p_b(k)\}$. Namely,

$$P_k[X = n] = \begin{cases} p_b(k) & 1 \leq n \leq 1/p_b(k), \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

By letting $\overline{X}(k)$ be the expected number of unmarked packets between two consecutive marked packets at slot k, $\overline{X}(k)$ is obtained as follows.

$$\begin{aligned} \overline{X}(k) &= \sum_{n=1}^{\infty} n \, P_k[X = n] \\ &= \frac{1/p_b(k) + 1}{2} \end{aligned} \quad (6)$$

We then derive state transition equations of the network model shown in Fig. 1. The state transition equations that we will derive hereafter become the basis of our steady state analysis. If all packets sent from the source host n are *not* marked at the RED gateway, corresponding ACK packets will be returned to the source host after the round-trip time. In this case, the congestion control mechanism of TCP increases the window size by one packet. On the contrary, if any of packets are discarded by the RED gateway, the congestion control mechanism of TCP throttles the window size to half. Here, we assume that all packet losses are detectable by duplicate ACKs [12]; that is, the number of discarded packets in a slot is assumed to be less than three. The probability that at least one packet is discarded from $w_n(k)$ packets is given by

$$\frac{w(k)}{1/p_b(k)} = w(k) \, p_b(k). \quad (7)$$

Therefore, the window size at slot $k + 1$ is given by

$$w(k+1) = \begin{cases} \frac{w(k)}{2} & \text{w. prob. } w(k) \, p_b(k) \\ w(k) + 1 & \text{otherwise} \end{cases} \quad (8)$$

According to our assumption, all packets that have sent in slot k are to be acknowledged till the beginning of slot $k+1$. So the current queue length at slot $k+1$ is given by

$$q(k+1) \quad = \quad \sum_{n=1}^{N} w_n(k) - B\left(\tau + \frac{q(k)}{B}\right). \quad (9)$$

Note that the first term of the right hand side is the total number of incoming packets at the RED gateway, and the second term is the number of outgoing packets. Note also that τ is the two-way propagation delay and $q(k)/B$ is the queueing delay in the buffer.

We then focus on the relation between $\overline{q}(k)$ and $\overline{q}(k+1)$. As explained in Section 2, the RED gateway updates the average queue length according to Eq. (1) at every packet arrival. Recalling that the current queue length $q(k)$ is assumed to be fixed during slot k, $\overline{q}(k)$ is obtained as

$$\begin{aligned}\overline{q}(k+1) \quad &= \quad (1-w_q)^{\sum_{n=1}^{N} w_n(k)}\overline{q}(k) \\ &+ \frac{w_q\{1-(1-w_q)^{\sum_{n=1}^{N} w_n(k)}\}}{1-(1-w_q)}\, q(k). \qquad (10)\end{aligned}$$

The network model shown in Fig. 1 is fully described by state transition equations given by Eqs. (8)–(10), and the state vector of the network, $\mathbf{x}(k)$, is given by

$$\mathbf{x}(k) = \begin{bmatrix} w_1(k) \\ \vdots \\ w_N(k) \\ q(k) \\ \overline{q}(k) \end{bmatrix}. \qquad (11)$$

3.2 Derivation of Averaged State Transition Equations

The state transition equations obtained in the previous section contain a probability, because the RED gateway marks each arriving packet in a probabilistic manner. For determining the behavior of the RED gateway in steady state, we introduce *averaged state transition equations*, which represent the typical behavior of TCP connections and the RED gateway.

We introduce a *sequence*, which is a series of adjacent slots in which all packets have not been marked by the RED gateway. A typical evolution of the window size within a sequence is illustrated by Fig. 3. Note that this figure depicts the case where the RED gateway discards one or more packets in slot $k-1$. The window size $w_n(k)$ is changed according to Eq. (8). Namely, until any packets from the source host are discarded by the RED gateway, the window size is increased by one packet at every slot. If one or more packets are discarded by the RED gateway, the window size is decreased to half. Such a process will be repeated indefinitely because of the congestion control mechanism of TCP.

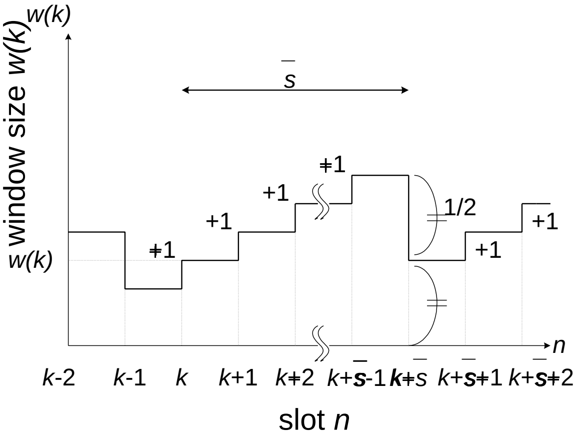


Figure 3: A typical evolution of the window size in a sequence.

The key idea for the next step is to treat the entire network as a discrete-time system where a time slot corresponds to a sequence that is composed of several time slots. Let $\overline{s}$ be the average number of slots that a sequence consists of. Then, $\overline{s}$ is the largest integer that satisfies the following inequality:

$$\overline{X}(k) \quad \leq \quad \sum_{i=0}^{\overline{s}-1}\sum_{n=1}^{N} w_n(k+i). \qquad (12)$$

In what follows, due to space limitation, we only show the case where window sizes of all source hosts are synchronized; that is, the window size of the source host n is equally given by $w(k)$. However, our steady state analysis presented below can be easily applied to the case where window sizes of all source hosts are not identical. When window sizes of all source hosts are identical, the above inequality is rewritten as follows.

$$\begin{aligned}\overline{X}(k) \quad &\leq \quad \sum_{i=0}^{\overline{s}-1} N\, w(k+i) \\ &= \quad N\left\{\overline{s}\, w(k) + \frac{\overline{s}(\overline{s}-1)}{2}\right\} \qquad (13)\end{aligned}$$

Solving this inequality for $\overline{s}$ yields

$$\begin{aligned}\overline{s} \quad = \quad &\left\lceil \frac{1}{2}\{1-2w(k)\right. \\ &\left. + \frac{\sqrt{N^2(1-2w(k))^2 + 8N\overline{X}(k)}}{N}\right\}\right\rceil. \quad (14)\end{aligned}$$

The averaged state transition equation for $w(k)$ between two consecutive sequences (i.e., slots from k to $k+\overline{s}$) is obtained from Eq. (8):

$$\begin{aligned} w(k+\overline{s}) &= \frac{w(k+\overline{s}-1)}{2} \\ &= \frac{w(k)+\overline{s}-1}{2}. \end{aligned} \tag{15}$$

Similarly, the averaged state transition equation for $q(k)$ between two consecutive sequences is obtained from Eq. (9):

$$\begin{aligned} q(k+\overline{s}) &= N\,w(k+\overline{s}-1) - B\left(\tau + \frac{q(k+\overline{s}-1)}{B}\right) \\ &\simeq \frac{N\,w(k+\overline{s}-1) - B\,\tau}{2} \\ &= N\,w(k+\overline{s}) - \frac{B\,\tau}{2}. \end{aligned} \tag{16}$$

In the above equation, $q(k+\overline{s}-1)$ is approximated by $q(k+\overline{s})$. Note that the difference between $q(k+\overline{s}-1)$ and $q(k+\overline{s})$ is upper-bounded by the number of TCP connections, N.

Then, we derive the averaged state transition equation for $\overline{q}(k)$ between two consecutive sequences. Using Eq. (10), $\overline{q}(k+\overline{s})$ is given by $w(k+\overline{s}-1)$, $q(k+\overline{s}-1)$ and $\overline{q}(k+\overline{s}-1)$ as

$$\begin{aligned} \overline{q}(k+\overline{s}) &= (1-w_q)^{N\,w(k+\overline{s}-1)}\overline{q}(k+\overline{s}-1) \\ &+ \frac{w_q\{1-(1-w_q)^{N\,w(k+\overline{s}-1)}\}}{1-(1-w_q)} q(k+\overline{s}-1). \end{aligned} \tag{17}$$

Recall that $\overline{X}$ is the average number of unmarked packets between two consecutive marked packets, i.e., the average number of unmarked packets in a sequence. By assuming that the current queue length does not change excessively, $\overline{q}(k+\overline{s})$ is approximated as

$$\begin{aligned} \overline{q}(k+\overline{s}) &\simeq (1-w_q)^{\overline{X}(k)}\overline{q}(k) \\ &+ \frac{w_q\{1-(1-w_q)^{\overline{X}(k)}\}}{1-(1-w_q)} q(k). \end{aligned} \tag{18}$$

The averaged state transition equations given by Eqs. (15), (16), and (18) describe the average behaviors of the window size, the current queue length, and the average queue length, respectively.

3.3 Derivation of Averaged Fixed Points

Because of the nature of TCP's congestion control mechanism, the window size of the source host oscillates indefinitely and is never converged to a fixed point. In what follows, we therefore derive *averaged fixed points*, which are defined as expected values in steady state, to understand the typical behavior of TCP connections and the RED gateway. Let w^*, q^*, and $\overline{q}^*$ be the averaged fixed points of the window size $w(k)$, the current queue length $q(k)$, and the average queue length $\overline{q}(k)$, respectively. The averaged fixed points can be easily obtained from Eqs. (15), (16), and (18) by equating $w(k) = w(k+\overline{s})$ and so on.

$$\begin{aligned} w^* &= \sqrt{\frac{1}{4} + \frac{1}{3N}\left(\frac{max_{th} - min_{th}}{max_p(\overline{q}^* - min_{th})} + 1\right)} \\ &\quad -\frac{1}{2} \end{aligned} \tag{19}$$

$$q^* = N\,w^* - \frac{B\,\tau}{2} \tag{20}$$

$$\overline{q}^* = q^* \tag{21}$$

Note that w^* does not mean the average window size in steady state, but it represents the expected value of the *minimum* window size. Instead, the average window size is obtained from w^* as follows.

$$\lim_{k\to\infty} \frac{\sum_{i=0}^{\overline{s}} w(k+i)\left(\tau + \frac{q(k+i)}{B}\right)}{\sum_{i=0}^{\overline{s}}\left(\tau + \frac{q(k+i)}{B}\right)} \simeq \frac{3w^*}{2} - 1 \tag{22}$$

In the above equation, $q(k+i)$ is approximated by $q(k)$ for $0 \le i \le \overline{s}$. Note that solving Eqs. (19) and (20) can give closed-form solutions of w^* and q^*, but closed-form solutions of w^* and q^* are too long to be included here.

4 Numerical Examples

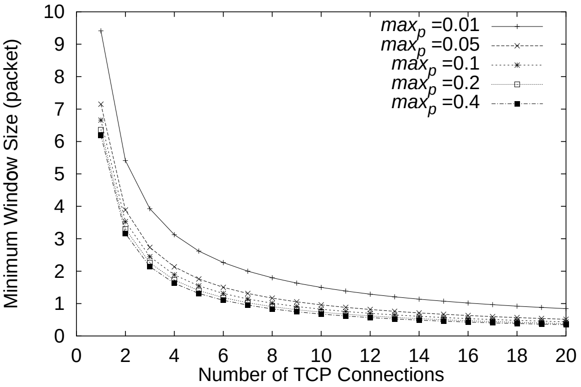


Figure 4: Minimum window size for different number of TCP connections ($B = 2$ [packet/ms], $\tau = 1$ [ms]).

Equations (19)– (21) indicate several interesting characteristics of the RED gateway. For instance, Eqs. (19) and (20) suggest that the window size of the source host and the queue length decreases as the number of TCP connections N or the maximum packet marking probability max_p increases. Such a tendency is clearly illustrated in

Figs. 4 and 5, where the averaged fixed points (i.e., expected minimum values) of the window size w^* and the queue length q^* are plotted, respectively. We use the following network parameters: the processing speed of the RED gateway $B = 2$ [packet/ms] and the two-way propagation delay $\tau = 1$ [ms], while the number of TCP connections N is varied from 1 to 20. The maximum packet marking probability max_p is also varied from 0.001 to 0.4, while other control parameters of the RED gateway are set to the values listed in Table 1. These figures indicate that the window size is heavily dependent on the number of TCP connections N, and that the queue length is mostly determined by the control parameter max_p.

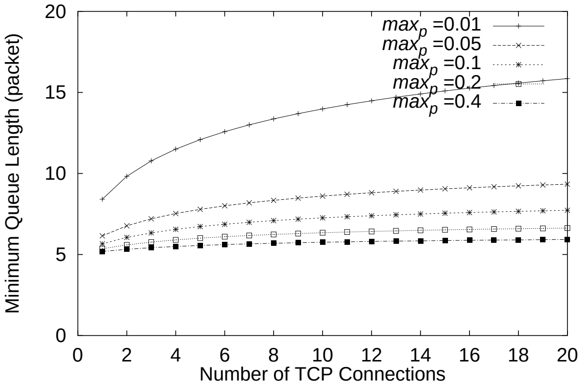
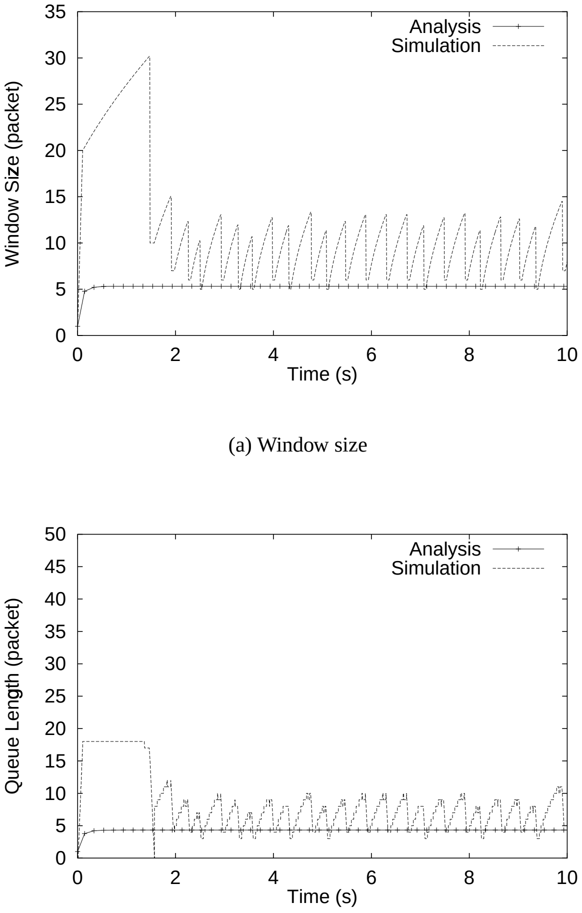


Figure 5: Minimum queue length for different number of TCP connections ($B = 2$ [packet/ms], $\tau = 1$ [ms]).

In what follows, we present a couple of simulation results to demonstrate the validity of our analysis since we have made several assumptions. We run simulation experiments for the same network model given by Fig. 1 using a network simulator *ns* [13]. We use the following network parameters in simulation experiments: the processing speed of the RED gateway $B = 2$ [packet/ms] (about 1.5 Mbit/s for the packet size of 1,000 bytes) and the two-way propagation delay $\tau = 1$ [ms]. The number of TCP connections N is set at either 1 or 5. For RED control parameters, the recommended set of control parameters listed in Table 1 are used.

Evolutions of the window size of the source host and the current queue length at the RED gateway obtained from simulation experiments are plotted in Figs. 6 and 7 for $N = 1$ and $N = 5$, respectively. In these figures, the averaged window size $w(k)$ and the averaged queue length $q(k)$ (i.e., minimum values in each sequence) are also plotted. Note that $w(k)$ and $q(k)$ are numerically computed from Eqs. (15), (16) and (18). One can find from these figures that our analytic results match the minimum values of the window size and the current queue length, indicating a close agreement of our analytic results

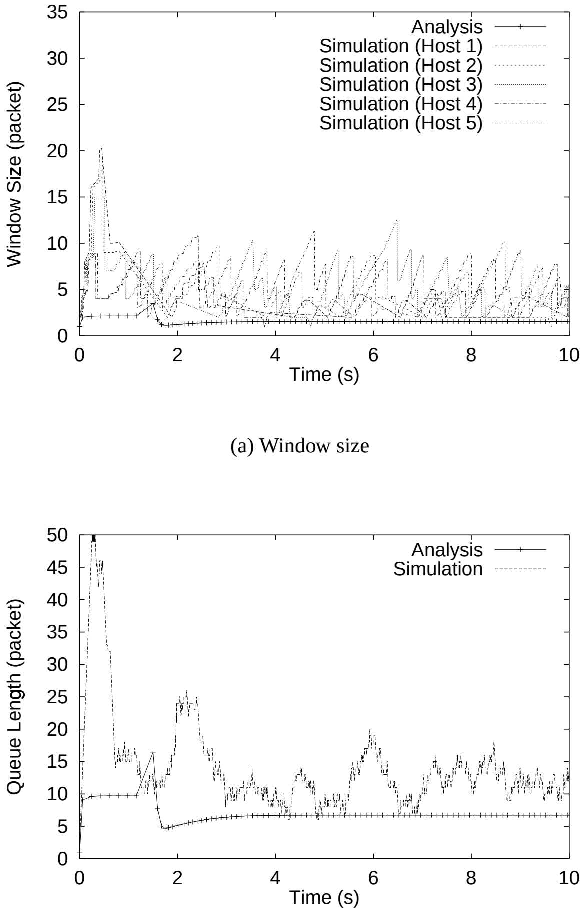


(a) Window size

(b) Queue length

Figure 6: Case of a single TCP connection ($B = 2$ [packet/ms], $\tau = 1$ [ms], $N = 1$).

with simulation ones.



(a) Window size

(b) Queue length

Figure 7: Case of 5 TCP connections ($B = 2$ [packet/ms], $\tau = 1$ [ms], $N = 5$).

References

[1] V. Jacobson, “Congestion avdoidance and control,” in *Proceedings of SIGCOMM ’88*, pp. 314–329, August 1988.

[2] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. New York: Addison-Wesley, 1994.

[3] E. Hashem, “Analysis of random drop for gateway congestion control,” *Technical Report MIT-LCS-TR-465*, 1989.

[4] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, August 1993.

[5] B. Barden et al., “Recommendations on queue management and congestion avoidance in the Internet,” *Request for Comments (RFC) 2309*, April 1998.

[6] D. Lin and R. Morris, “Dynamics of random early detection,” in *Proceedings of ACM SIGCOMM ’97*, September 1997.

[7] M. May, J. Bolot, C. Diot, and B. Lyles, “Reasons not to deploy RED,” in *Proceedings of IWQoS ’99*, pp. 260–262, March 1999.

[8] W. chang Feng, D. D. Kandlur, D. Saha, and K. G. Shin, “A self-configuring RED gateway,” in *Proceedings of IEEE INFOCOM ’99*, March 1999.

[9] T. J. Ott, T. V. Lakshman, and L. Wong, “SRED: Stabilized RED,” in *Proceedings of IEEE INFOCOM ’99*, March 1999.

[10] M. May, T. Bonald, and J.-C. Bolot, “Analytic evaluation of RED performance,” to be presented at *IEEE INFOCOM 2000*, March 2000.

[11] V. Sharma, J. Virtamo, and P. Lassila, “Performance analysis of the random early detection algorithm,” available at *http://keskus.tct.hut.fi/tutkimus/com2/publ/redanalysis.ps*, September 1999.

[12] W. R. Stevens, “TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms,” *Request for Comments (RFC) 2001*, January 1997.

[13] “LBNL network simulator (ns).” available at http://www-nrg.ee.lbl.gov/ns/.